A2 concld-

indicates some condition or action to be taken. For example, the messages might indicate a new network server has entered the ring network. Similarly, each of the client requests is represented by one or more of the packets 1004. Some packets include a self-identifying heartbeat message. As long as the heartbeat message circulates, the ring network is assumed to be free of faults. In the system 100, a token is implicit in that the token is the lower layer packet 1004 carrying the heartbeat message. Receipt of the heartbeat message indicates that the nearest transmitting ring member is functioning properly. By extension, if the packet 1004 containing the heartbeat message can be sent to all ring members, all nearest receiving ring members are functioning properly and therefore the ring network is fault-free.

Replace the paragraph beginning on page 24, line 4 with the following rewritten paragraph:

A3

An exemplary embodiment of the system 100 is described below. Each client is an Intel Pentium II 266 with 64 or 128 megabytes (MB) of random access memory (RAM) running Red Hat Linux 5.2 with version 2.2.10 of the Linux kernel. Each network server is an AMD K6-2 400 with 128 MB of RAM running Red Hat Linux 5.2 with version 2.2.10 of the Linux kernel. The dispatch server is either a server similar to the network servers or a Pentium 133 with 32 MB of RAM and a similar software configuration. All the clients have ZNYX 346 100 megabits per second Ethernet cards. The network servers and the dispatch server have Intel EtherExpress Pro/100 interfaces. All servers have a dedicated switch port on a Cisco 2900 XL Ethernet switch. Appendix A contains a summary of the performance of this exemplary embodiment under varying conditions.

Delete Appendix A at pages 27-34.

Replace the subtitle beginning at page 35, line 1 with the following rewritten subtitle:

Appendix A

Replace the paragraph beginning at page 35, line 5 with the following rewritten paragraph:

A4 Our results demonstrate that in tests of real-world (and some not-so-real-world) scenarios, our SASHA architecture provides a high level of fault tolerance. In some cases, faults might go unnoticed by users since they are detected and masked before they make a significant impact on the level of service. Our fault-tolerance experiments are structured around three levels of service requested by client browsers: 2500 connections per second (cps), 1500 cps, and 500 cps. At each requested level of service, we measured performance for the following fault scenarios: no-faults, a dispatcher server faults, three server faults, and four server faults. Figure 1A summarizes the actual level of service provided *during the fault detection and recovery interval* for each of the failure modes. In each fault scenario, the final level of service was higher than the level of service provided during the detection and recovery process. The rest of this section details these experiments as well as the final level of service provided after fault recovery.

N.E. Delete the caption following the figure at page 35.

Replace the paragraph beginning at page 36, line 2 with the following rewritten paragraph:

A5 In the first case, we examined the behavior of a cluster consisting of five server nodes and the K6-2 400 dispatcher. Each of our five clients generated 500 requests per second. This was the maximum sustainable load for our clients and servers, though dispatcher utilization suggests that it may be capable of supporting up to 3,300 connections per second. Each test ran for a total of 30 seconds. This short duration allows us to more easily discern the effects of node failure. Figure 1A shows that in the base, non-faulty, case we are capable of servicing 2,465 connections per second.

N.E. Delete the caption following the figure at page 38.

Replace the paragraph beginning at page 38, line 15 with the following rewritten paragraph:

A6 Cmt As we see in Figure 2A, at 500 and 1,000 cps, we are capable of servicing all the requests. By the time we reach 1,500 cps, we can service just over 1,000. 2,000 and 2,500 cps actually see worse service as the dispatcher becomes congested and packets are dropped, nodes must retransmit, and traffic flows less smoothly. The 1996 games saw, at peak load, 600 cps. That is, our capacity to serve is 1.8 times the actual peak load. In similar fashion, we believe our 1998

A6 Cncld.

vintage hardware is capable of dispatching approximately 3,300 connections per second, again about 1.8 times the actual peak load. While we only have two data points from which to extrapolate, we conjecture that COTS systems will continue to provide performance sufficient to service even the most extreme loads easily.